

# Compiler Lec

~~Strings~~

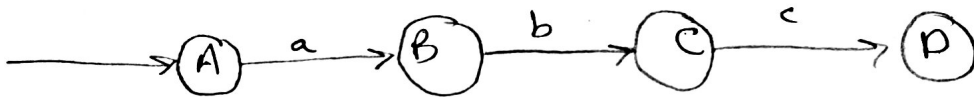
Epsilon move

— يمكن نقله A إلى c ~ ~ ~ (Consuming) للمبار.

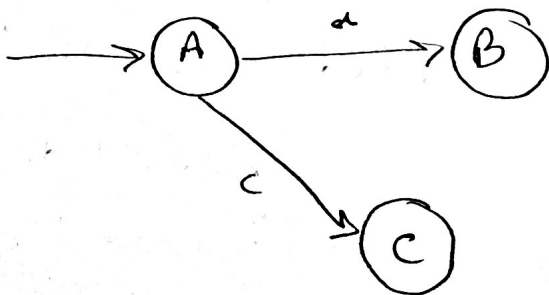
— أعطى اختياراً كثيرة.

— يمكن تحركه (input) و (input string) يساراً  
مختلفة (كل المراه تؤدي إلى ردود).

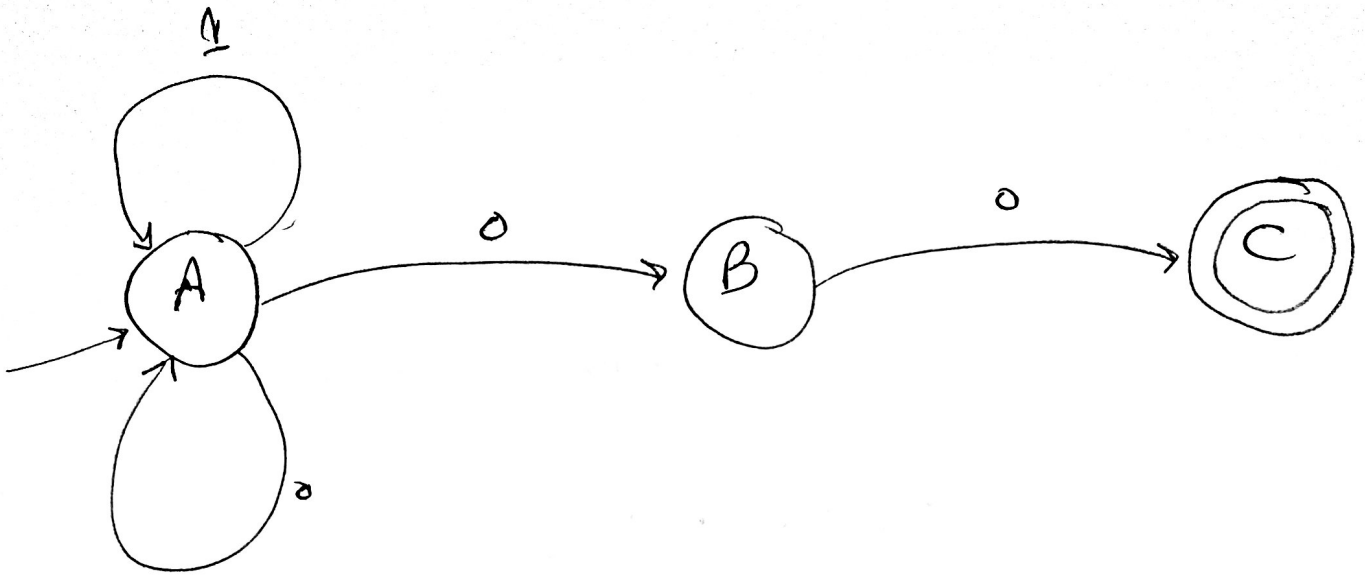
abc



Deterministic  
Finite  
Automata.



Non Deterministic  
Finite  
Automata.



→ Non Deterministic .  
 ← عشا بعد ( 2 transition ) لا 0 1 لبداية .

Note Input : 1 0 0  
 state : ~~{A}~~ {A} {A,B} {A,B,c}

نظروا بين قوسين ممكن  
 multiple choises بسبب

← لو آخر state فيها (accept state) اذا

الحالة دي Accepted .

Rejected

← لو وصلت لنهاية string (Accept state)

← لو لم اهدل حتى نهاية ال string (state) حتى لو (Accepted)

RE ← يعرف على طريقه ال (operation)

~~more~~ DFA ← يرمز (diagrams) لـ (states) ، ~~no choices~~

NDFA ← " " " " (more ~~to~~ choices to consider)

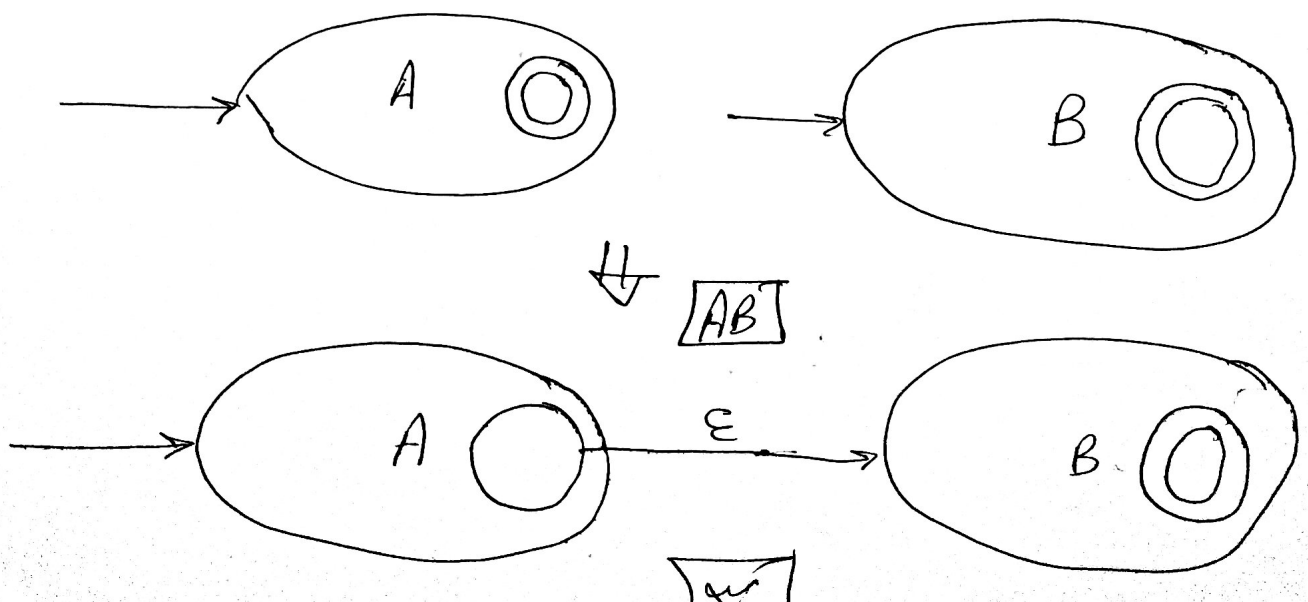
← اللغة - يوفونوا نفس ال (formal language)

DFA ← المسار واحد لا يتغير .

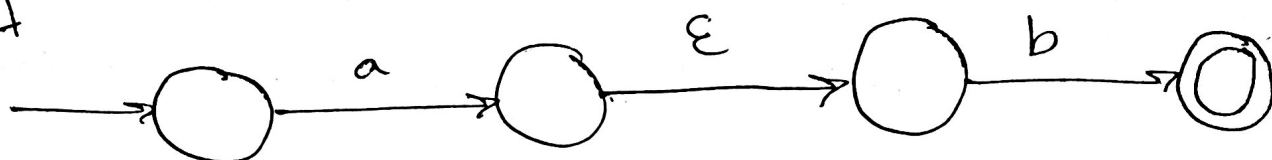
RE مختلفة ممكنه اعمل فيها DFA مختلفة بس

لو عملت (analyze) ليهم هيكملعولي نفس النتيجة .

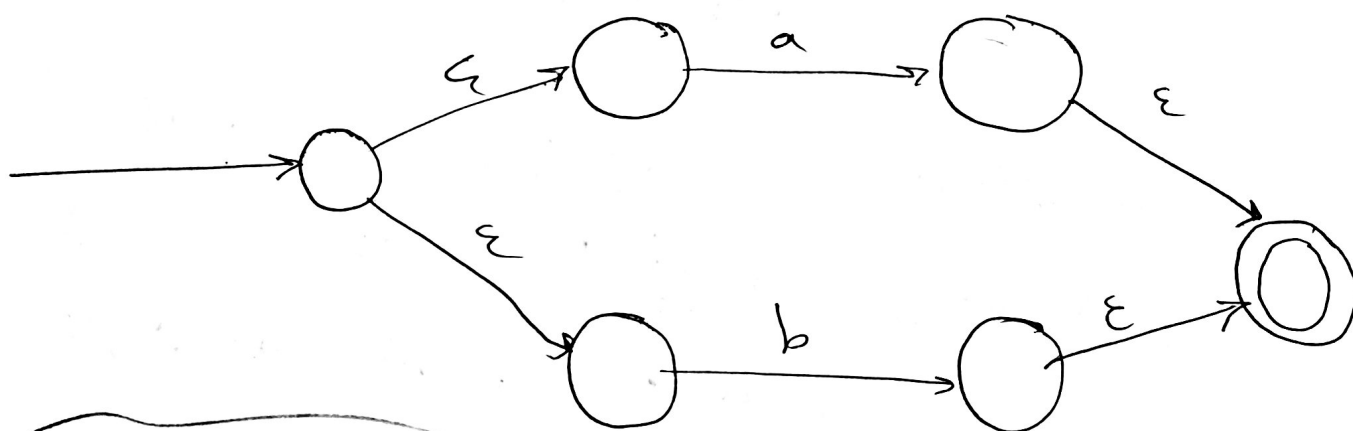
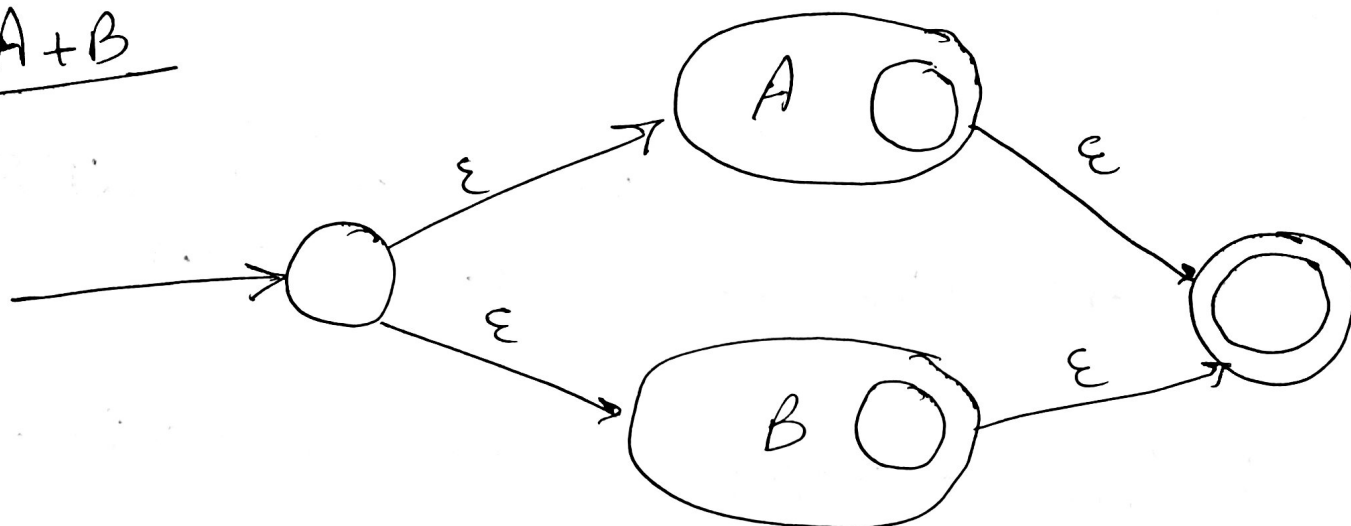
لكر لو عملت (NDFA) ~~هيكملعولي~~ هيبقى الحل واحد .  
'A' 'B'



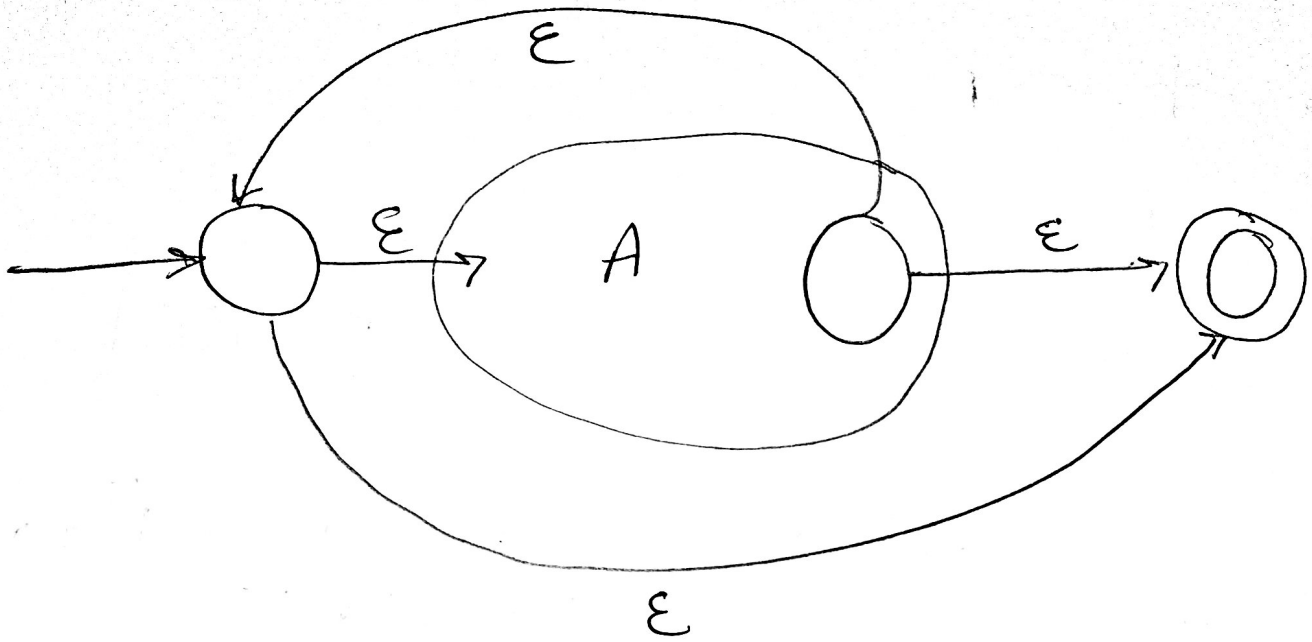
AB



A+B

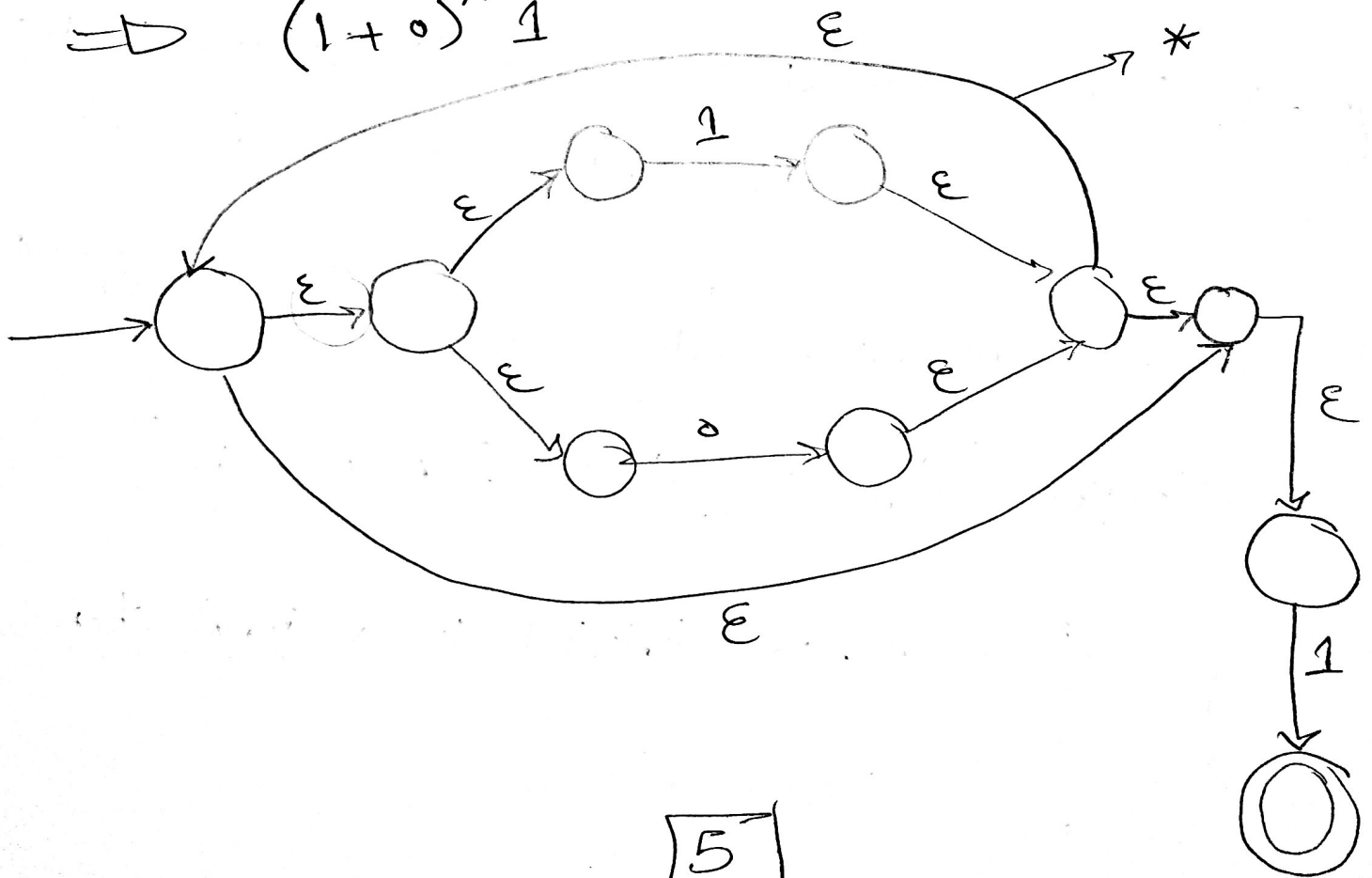


9



→ RE to NFA ( $\epsilon$ )

$\Rightarrow (1+0)^* 1$



5



$a(X)$   $\rightarrow$  set of states

$$\epsilon\text{-closure}\{G\} = \{G, H, A, I, A, B, C, D\}$$

$\downarrow$   
 $X$

وكل حرف منظم ليس  $y$  or  $x$  (small character)

$$a(X) = \{y \mid x \in X_n, x \xrightarrow{a} y\}$$

$$= Y$$

$\epsilon\text{-closure}\{Y\}$   $\leftarrow Y$  (نفسه)  $\rightarrow$   $Y$  (نفسه)

start state  $\rightarrow \epsilon\text{-closure}(a$

(data)  $\rightarrow$  input ch.  $\rightarrow$  (consuming)  $\rightarrow$   $\rightarrow$

(c

(d

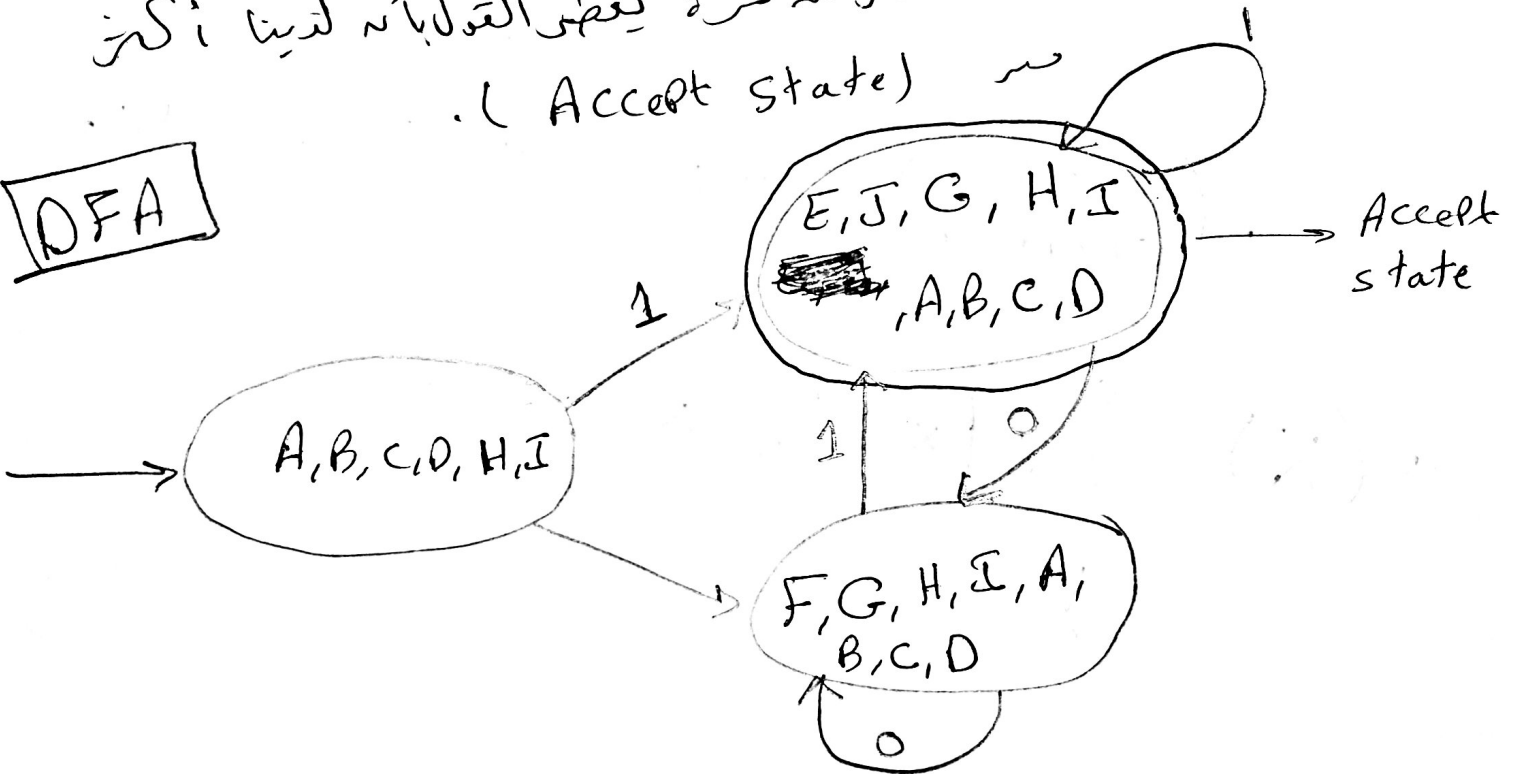
1)

$$\epsilon\text{-closure}\{A\} = \{A, B, C, D, H, I\}$$

A ← مش هتظهر غير هنا .

لأنه ظاهرها أكثر مرة يعطى القبول لدينا أكثر  
مر (Accept state)

DFA



$$\textcircled{1}(X) = \{E, J\}$$

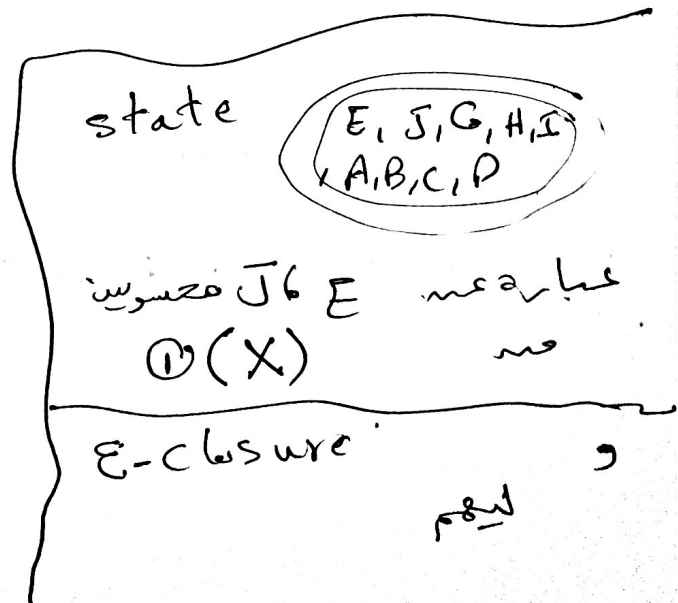
$$C \xrightarrow{1} E$$

$$I \xrightarrow{1} J$$

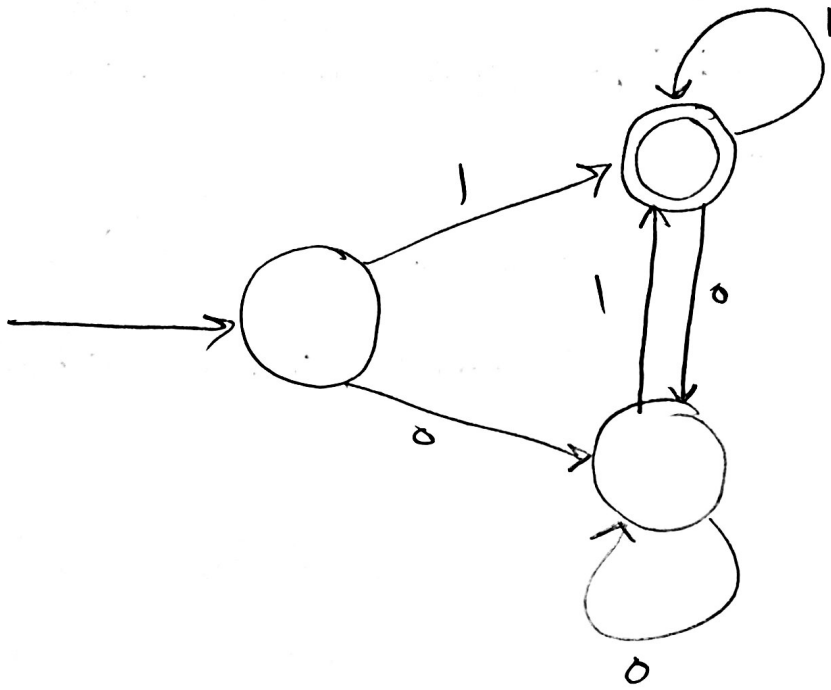
$$\epsilon\text{-closure}\{\textcircled{1}(X)\}$$

$$\epsilon\text{-closure}\{E, J\}$$

⊆



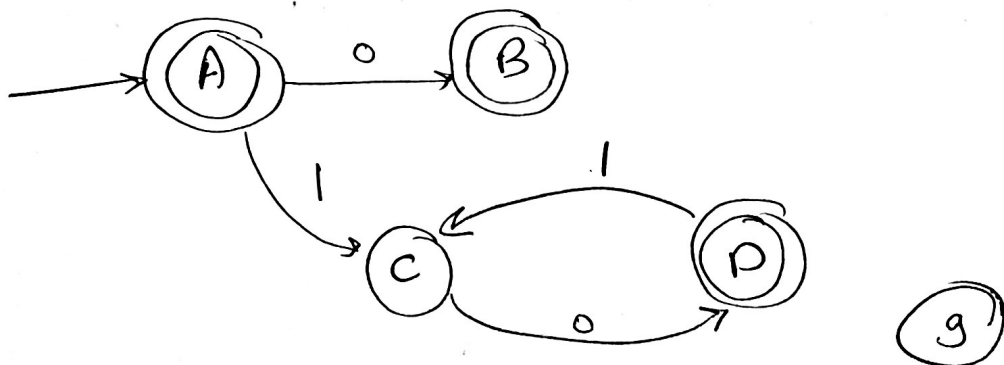
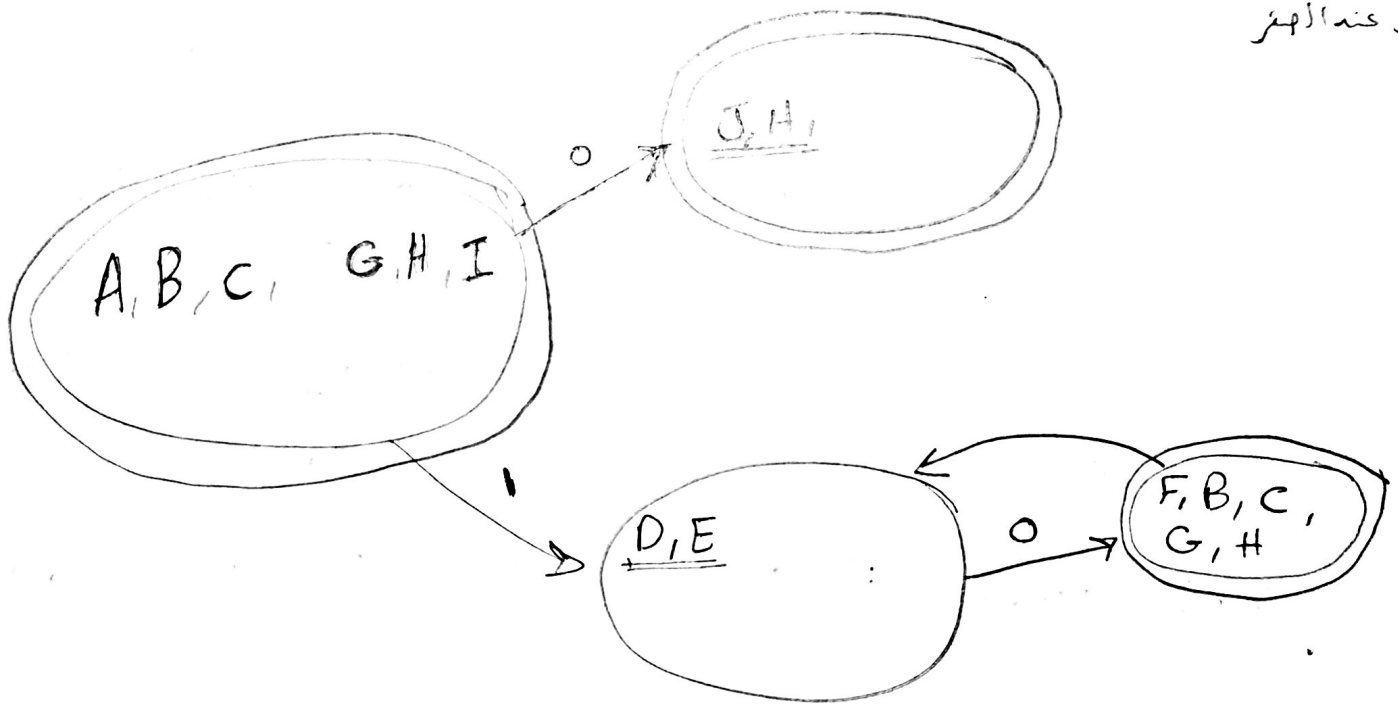




DFA

$A = \{ A, B, C, E, G, H, I \}$

ج عندا ہر



→ To implement DFA you may use table.

مثلاً في آخر مرة ممكن  
Columns (input choices)

rows  
(state)

	0	1
A	B	C
B		
C	D	
D		C

ممكن لو عايز تملأ الفراغ  
دي بتكتب (state)  
لو حدها بحيث لو عندك حالة  
ممكن تتش لأى حالة أخرى  
تكتب ال (state)  
دي

(optional) مش إجباري عليه.

implementation  
i = 0 ; (index) for input char:  
state = 0 ; start state

while (input[i])

طالعني (i/p) متحرك زي  
ما i

{

state = T[state, input[i]]

{مفروض دول بيتحرك  
next state}

}

++i غلط